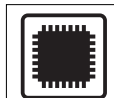
続

実設計に応用できる演算回路スキルを身につけよう

デバイスの記事

外村元伸

第4回 3次元空間モデルを扱う応用分野に現れる除算・開平・逆数・逆開平数とその演算回路設計

みなさんおなじみの昨今のゲーム機では，3次元コンピュータ・グラフィックス技術があたりまえのように使われています．当然，演算処理能力がかなり要求されます．この要求に応えるために使われる演算方式の基本となることについて解説していきます．どのような場面にどの演算が要求されているのかを把握してもらうために，3次元空間モデルの扱い方に触れてから，それらに必要な演算とそれらの設計法に関することを掘り下げていきます．（筆者）

3次元コンピュータ・グラフィックスやコンピュータ・ビジョン（立体画像認識）の分野では，3次元空間モデルを扱います．**図1**に示すような，ベクトルで表現する座標系の知識が必要です．特に，3次元座標系0-XYZにおいて，法線ベクトル**n**(x, y, z)を回転軸とする半径rのベクトルの回転θがよく出てきます．そして，法線ベクトル**n**(x, y, z)を単位長（長さ1）のベクトルにすること，つまり正規化することが要求されます．式で表現すれば，次のように変換することです．

$$(x,\ y,\ z) \xrightarrow{\text{正規化}} \left(\frac{x}{\sqrt{x^2+y^2+z^2}},\ \frac{y}{\sqrt{x^2+y^2+z^2}},\ \frac{z}{\sqrt{x^2+y^2+z^2}} \right) \cdots (1)$$

● 魚眼円形画像を平面画像に補正変換する

このような式が実際に現れる場面の例として，魚眼画像の補正変換を簡単に紹介します．詳しくは今後の連載の中で解説しますが，まずはどのような原理によって扱っているのかを概観します．

写真1に魚眼レンズで撮影した画像を示します．円形にひずんだ画像になっていることが特徴です．円形にひずんだ魚眼画像は，メカニカルな動作部なしに，視野角180°を越える全方位画像を一度に収めることができます．そして，円形魚眼画像の任意の部分を切り出し，平面化の補正変換を行えば，通常と違和感のない画像が得られます．

では，このような補正変換はどのようにすれば行えるのでしょうか．画像のゆがみ具合を測定して変換するキャリブレーションと呼ばれる補正も行われますが，ここではもっと簡単な仮想球面モデルを使って変換する方法を採り上

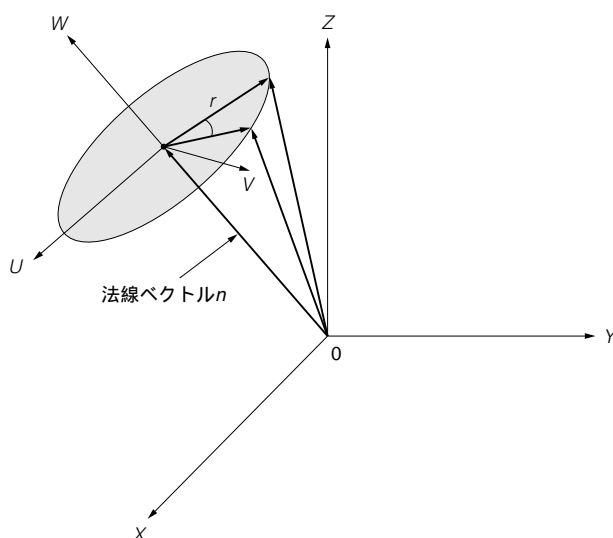



図1　3次元ベクトル空間での0-XYZ座標系と回転
3次元空間モデルを扱うために，3次元ベクトルによって座標を表現する．また，法線ベクトルnを軸とする3次元の回転が頻繁に現れる．

KeyWord 3次元コンピュータ・グラフィックス，3次元コンピュータ・ビジョン，立体画像認識，魚眼円形画像，魚眼レンズ，平面化補正変換，除算，開平，逆数，逆開平数演算，on-the-fly変換

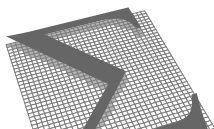



写真1
魚眼レンズで撮影した円形にゆがんだ画像
写真左の魚眼レンズで撮影した円形にゆがんだ画像の一部を切り出して，写真右の通常のレンズで撮影した写真のように見えるように，魚眼円形画像に対して平面化の補正変換を行うことが目的．

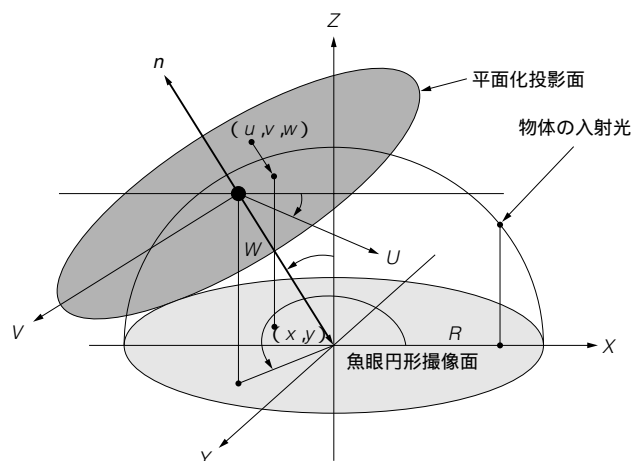
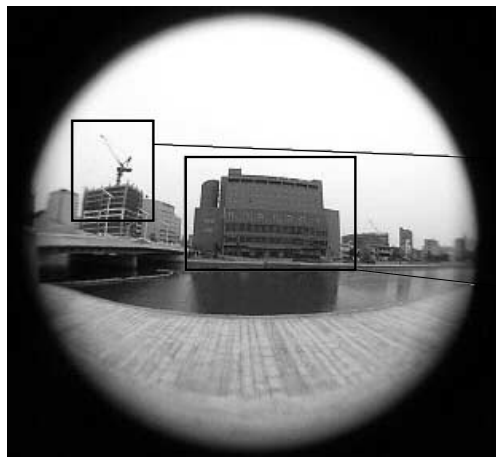



図2　魚眼円形画像を平面化補正変換する幾何学モデル
魚眼円形撮像面に対して，仮想球面を設けることで，魚眼レンズのはたらきを簡単にモデル化する．円形にゆがんだ画像を平面化するために，仮想球面に接する平面が設けられ，その平面上に仮想球面を介して円形像が投影される．

げます．魚眼レンズの画像取得の仕組みを簡単にモデル化すると，**図2**に示すようになります．魚眼を仮想球面で簡易モデル化します．そうすると，撮影物体から飛来した光は，仮想球面に垂直に入射し，撮像面に相当する半径Rの円形の平面（魚眼円形撮像面）に垂直に下りたところに像を造ります．これで視野角180 °の魚眼レンズのモデル化ができます．あとはこの幾何学モデルに従って，平面化変換する式を導きます．平面化は仮想球面に接する平面を設定し，その平面に投影される像を求めることによって行います．投影平面が仮想球面に接する点の座標位置は，XYZ座標系によって接点を通る法線ベクトル$\boldsymbol{n}(x, y, z)$を決めることで指定します．魚眼円形撮像面上で，X軸と法線ベクトル$\boldsymbol{n}$の投影線がなす角αを方位角と呼びます．それから，Z軸と法線ベクトル$\boldsymbol{n}$のなす角βは天頂角と呼びます．平面化投影面はUVW座標系で表し，X軸とU軸のなす角をϕとします．角(α, β, ϕ)によって法線ベクトル$\boldsymbol{n}(x, y, z)$の座標位置が指定されます．

● 平面化補正変換の基本的な形

さて，平面化補正変換する式を次に示しますが，今回は演算式の基本的な形を知ることが目的なので，結果だけを示します．詳しい導出の仕方については，これからの連載記事の中で紹介します．UVW座標系で求めた平面化投影面上の位置(u, v, w)における画素値を求めるために，まず，UVW座標系からXYZ座標系へ座標変換する式を得ます．その式は，

$$\begin{cases} x = \dfrac{R\left(uA - vB + w\sin\beta\sin\alpha\right)}{\sqrt{u^2+v^2+w^2}} \\ y = \dfrac{R\left(uC - vD - w\sin\beta\cos\alpha\right)}{\sqrt{u^2+v^2+w^2}} \end{cases} \quad \cdots\cdots (2)$$

です．ここで，(x, y)は魚眼円形撮像面での座標位置，Rは魚眼円形像の周辺半径で，方位角α，天頂角β，X軸とU軸のなす角ϕによって座標の回転が，

$$\begin{cases} A = \cos\phi\cos\alpha - \sin\phi\sin\alpha\cos\beta \\ B = \sin\phi\cos\alpha + \cos\phi\sin\alpha\cos\beta \\ C = \cos\phi\sin\alpha + \sin\phi\cos\alpha\cos\beta \\ D = \sin\phi\sin\alpha - \cos\phi\cos\alpha\cos\beta \end{cases} \quad \cdots\cdots (3)$$

で表されます．これで，平面化投影面位置(u, v, w)の魚眼円形撮像面での対応位置(x, y)が求まるので，位置(x, y)に相当する画素値は，画像の適当な補間によって求めること

ができます．このようにして魚眼円形像を平面化画像に補正変換します．式(2)において，

$$\frac{1}{\sqrt{u^2+v^2+w^2}} \quad \cdots\cdots(4)$$

は，ベクトルの正規化を行っているところです．

● 3次元空間データ(座標)を扱うための基本演算

式(1)や式(2)で示したように，

$$f=\frac{t}{\sqrt{x^2+y^2+z^2}} \quad \cdots\cdots(5)$$

という形の演算が，3次元空間座標を扱う場合に要求されていることが分かります．この中には，2乗，開平，除算あるいは2乗，開平，逆数，乗算という演算が含まれています．また，2乗，逆開平数と乗算によって求めるということも考えられます．

さて，この中で実現しなければならない除算，開平，逆数と逆開平数について，どのように考えていけばよいか，これから解説します[注1]．実現の方法は，いろいろ提案されています．そこで，大きく分けて2種類の方法，①1ビットずつ逐次的に値を求めていく方法(ハードウェア量は少ないが演算時間がかかる)，②並列に値を求める方法(ハードウェア量は多いが高速)について注目し，それぞれについて説明します．今回は①の逐次的な方法です．SRT(Sweeney, Robertson, Tocher)法と呼ばれている部類に入るものです．

● 除算・開平・逆数と逆開平数演算

除算，開平，逆数，逆開平演算は，それぞれ以下の式によって定義されます．

$$\begin{cases} \text{除算}: Q=\dfrac{Y}{X} \\ \text{開平}: Q=\sqrt{X} \\ \text{逆数}: Q=\dfrac{1}{X} \\ \text{逆開平数}: Q=\dfrac{1}{\sqrt{X}} \end{cases} \quad \cdots\cdots(6)$$

ここで，Yは被除数，Xは除数または被開平数，Qは商，開平値，逆数値または逆開平数値です．除算の場合は$\frac{1}{2} \leqq X$，$Y<1$，逆数の場合は$\frac{1}{2} \leqq X<1$，開平の場合は$\frac{1}{4} \leqq X<1$，逆開平数の場合は$\frac{1}{4}<X<1$と範囲を仮定します．範囲をこのように制限する理由は，一般的に演算アルゴリズムの扱いを簡単にするためです．また，浮動小数点演算の場合は，データの範囲を$\frac{1}{2} \leqq X<1$に制限して(正規化と呼ばれる)，仮数部を[0.1...]bのように2進数表現に保つためでもあります．

式(6)を，剰余Rを求める形にそれぞれ変形します．

$$\begin{cases} \text{除算}: R=Y-QX \\ \text{開平}: R=X-Q^2 \\ \text{逆数}: R=1-QX \\ \text{逆開平数}: R=1-Q^2X \end{cases} \quad \cdots\cdots(7)$$

$Q=\sum_{i=0}^{n} q_i \cdot 2^{-i}$とし，1ビットずつ$q_i$値を決定していく場合は，基数2と呼ばれます．次に，基数2の場合の漸化式の導き方を示します．iステップ目の部分剰余R_iを，

$$\begin{cases} \text{除算}: R_i=2^i\left(Y-Q_iX\right) \\ \text{開平}: R_i=2^i\left(X-Q_i^2\right) \\ \text{逆数}: R_i=2^i\left(1-Q_iX\right) \\ \text{逆開平数}: R_i=2^i\left(1-Q_i^2X\right) \end{cases} \quad \cdots\cdots(8)$$

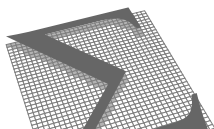
とします．まず，除算について以下のように漸化式が導けます．$Q_i=\sum_{k=0}^{i-1} q_k \cdot 2^{-k}$とすると，

$$\begin{aligned} R_{i+1} &= 2^{i+1}\left(Y-Q_{i+1}X\right) \\ &= 2\cdot 2^i\left(Y-Q_iX-q_i\cdot 2^{-i}X\right) \\ &= 2\left(R_i-q_iX\right) \end{aligned} \quad \cdots\cdots(9)$$

となります．開平については，$Q_i=\sum_{k=0}^{i} q_k \cdot 2^{-k}$として，

$$\begin{aligned} R_{i+1} &= 2^{i+1}\left(X-Q_{i+1}^2\right) \\ &= 2\cdot 2^i\left(X-\left(Q_i+q_{i+1}\cdot 2^{-(i+1)}\right)^2\right) \\ &= 2\cdot 2^i\left(\left(X-Q_i\right)-2q_{i+1}\cdot 2^{-(i+1)}Q_i-q_{i+1}^2\cdot 2^{-2(i+1)}\right) \\ &= 2\left(R_i-q_{i+1}\cdot Q_i-q_{i+1}^2\cdot 2^{-(i+2)}\right) \end{aligned} \quad \cdots\cdots(10)$$

となります．逆数については，除算の被除数Yが1であることから，$Q_i=\sum_{k=0}^{i-1} q_k \cdot 2^{-k}$として，

注1　式(3)の中で三角関数が使われているが，これについては別途解説する予定．あるいは，本誌2006年9月号[4]で少し触れているので参考のこと．

$$R_{i+1} = 2^{i+1}\left(1 - Q_{i+1}X\right)$$
$$= 2 \cdot 2^i\left(1 - Q_iX - q_i \cdot 2^{-i}X\right)$$
$$= 2\left(R_i - q_iX\right) \quad \cdots\cdots(11)$$

となります．逆開平数については，$Q_i = \sum_{k=0}^{i} q_k \cdot 2^{-k}$ として，

$$R_{i+1} = 2^{i+1}\left(1 - Q_{i+1}^2X\right)$$
$$= 2 \cdot 2^i\left(1 - \left(Q_i + q_{i+1} \cdot 2^{-(i+1)}\right)^2 X\right)$$
$$= 2 \cdot 2^i\left(\left(1 - Q_i^2\right) - 2q_{i+1} \cdot 2^{-(i+1)}Q_iX - q_{i+1}^2 \cdot 2^{-2(i+1)}X\right)$$
$$= 2\left(R_i - q_{i+1} \cdot Q_iX - q_{i+1}^2 \cdot 2^{-(i+2)}X\right) \quad \cdots\cdots(12)$$

となります．以上をまとめると，各演算アルゴリズムの第iステップ目の漸化式は，

$$\begin{cases} \text{除算：} R_{i+1} \leftarrow 2\left(R_i - q_i \cdot X\right),\ R_0 = Y \\ \text{開平：} R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot \left(Q_i + q_{i+1} \cdot 2^{-(i+2)}\right)\right), \\ \qquad Q_{i+1} = Q_i + q_{i+1} \cdot 2^{-(i+1)},\ R_0 = X,\ Q_0 = 0 \\ \text{逆数：} R_{i+1} \leftarrow 2\left(R_i - q_i \cdot X\right),\ R_0 = 1 \\ \text{逆開平数：} R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot X\left(Q_i + q_{i+1} \cdot 2^{-(i+2)}\right)\right) \\ \qquad Q_{i+1} = Q_i + q_{i+1} \cdot 2^{-(i+1)},\ R_0 = 1 - X, Q_0 = 1 \end{cases} \cdots(13)$$

となります．これらは筆算の原理でもあります．

● ディジット選択規則

さて，式(8)のそれぞれのq_iは，「商ディジット(除算)」，「開平ディジット」，「逆数ディジット」，「逆開平ディジット」と呼ぶことにします．これらを求めるには，各種ディジットを決定するための選択規則が必要です．式(13)の漸化式で求めた部分剰余R_iの演算結果から，ディジットq_iを決定します．ディジットq_iの選択集合をどのように設定するかを考えなければなりません．$q_i \in \{+1, -1\}$, $q_i \in \{+1, 0, -1\}$ などとする方法が提案されています．それから，この部分剰余R_iの演算には，キャリ伝播が伴います．完全にキャリ伝播を実行すると，けた数に比例した時間がかかってしまいます．

今回紹介するのは，式(13)の漸化式をできるだけはやく演算してディジットq_iを決定するための方法です．演算には，キャリ・セーブ形式(sum, carry)を使います．そして，ディジットq_i決定のために，上位mけた分だけをキャリ伝播させます．最もはやくディジットq_iを決定できる，つまりこの選択論理の段数が最も少ないけた数mはいくつなのかという興味深い問題があります．

ここではこの問題に直接触れずに，除算，開平，逆数と逆開平数で共通に採用できるディジット選択規則の例を**表1**に示します．ディジットの選択集合は，$q_i \in \{+1, 0, -1\}$ です．ディジットq_i決定のために上位3けた分(r_{-1}, r_0, r_1)だけをキャリ伝播させますが，さらに(r_{-1}, r_0, r_1) = (0, 0, 0)のときは，上位4けた目のキャリ・セーブ値(s_2, c_2)を使います．(s_2, c_2)がゼロでないときは，q_i = +1を選択します．(s_2, c_2)がゼロのときは，q_i = 0を選択します．‘ ＊ ’は，don ’t careの意味です．r_{-1}が2進数の2^1のけた，r_0が2^0のけた，そしてr_1が2^{-1}のけたです．また，r_{-1}が2の補数表現の符号部のけたです．(r_{-1}, r_0, r_1) = (1, 0, 0)の値になると選択規則の範囲からはみ出してしまいますが，初期値が(r_{-1}, r_0, r_1) = (0, 0, 1)なので，途中で(r_{-1}, r_0, r_1) = (1, 0, 0)の値になるようなことがないように規則が導かれています．

表1 ディジット選択規則(＊：don't care)

q_i	r_{-1}	r_0	r_1	$(s_2,\ c_2)$
+1	0	1	1	＊
+1	0	1	0	＊
+1	0	0	1	＊
+1	0	0	0	Not 0
0	0	0	0	0
0	1	1	1	＊
−1	1	1	0	＊
−1	1	0	1	＊
未定義	1	0	0	＊

● on-the-fly変換

ディジット集合が，$q_i \in \{1, 0, \bar{1}\}$ (連続してけた値表記しやすいように，$\bar{1}$ = −1とする)であるために，2進数{0, 1}に変換する必要があります．すべてのけたのディジットが求まってから，一気に変換する方法もありますが，ここでは，ディジットq_i が1ビットずつ求まることから，q_i が求まるごとに2進数表現に変換するon-the-flyと呼ばれる変換方法を紹介します．

on-the-fly変換では，最後のけたのディジットが決定されれば，即座に2進数表現による結果を得ることができます．on-the-fly変換の原理は，次のように考えるとわかりやすいでしょう．$q_i = \bar{1}$ が選択されたら，一つ上位のけたから1を借りてきて，つまり，$1\bar{1} \rightarrow 01$と変換するので，予備として，絶えず一つ少ない途中結果の数を保持しておけばよいのです．それで，途中結果を保持するための二つのレ

ジスタを用意します．選択されたディジットq_iによって決定される途中結果を格納するPレジスタを設けます．そして，Pレジスタの内容より一つ少ない途中結果を格納するNレジスタをもう一つ設けます．**図3**に示すように，$q_i=\bar{1}$が選択された場合は一つ少ない途中結果のNレジスタを使い，iけた目に1を設定し，Pレジスタの新しい内容とします．また，iけた目に0を設定し，Nレジスタの新しい内容とします．$q_i=0$が選択された場合は，途中結果のPレジスタを使い，iけた目に0を設定し，Pレジスタの新しい内容とします．これの一つ少ない数は－1する（$q_i=\bar{1}$ を選択したときの新しいPレジスタの設定と同じ）ことなので，途中結果のNレジスタを使い，iけた目に1を設定してNレジスタの新しい内容とします．$q_i=1$が選択された場合は，途中結果のPレジスタを使い，iけた目に1を設定してPレジスタの新しい内容とします．また，iけた目に0を設定し，Nレジスタの新しい内容とします．

一石二鳥のハードウェア，そして一石四鳥のハードウェアへ

さて，式(13)のアルゴリズムをハードウェア化することを考えます．まず，除算と開平の漸化式

$$\begin{cases} 除算：R_{i+1} \leftarrow 2\left(R_i - q_i \cdot \boxed{X}\right) \\ 開平：R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot \boxed{\left(Q_i + q_{i+1} \cdot 2^{-(i+2)}\right)}\right) \end{cases} \cdots(14)$$

の□で囲った部分に注目してください．除算の除数Xと開平の$(Q_i + q_{i+1}2^{-(i+2)})$とを共用して，一つのレジスタで扱えば，ほぼ同じ形なので，除算・開平演算器は共通に設計できるということになります．除算・開平共用演算器のブロック図は**図4**に示すようになります．除算の除数Xと開平の$(Q_i + q_{i+1} \cdot 2^{-(i+2)})$とを共用するレジスタXを設けます．部分剰余レジスタはキャリ・セーブ形式(s_i, c_i)です．ただし，上位3ビット分だけはキャリ伝播を実行し，2進数形式(r_{-1}, r_0, r_1)にします．そして，上位3けた(r_{-1}, r_0, r_1)と上位4けた目のキャリ・セーブ値(s_2, c_2)を見て，**表1**のディジット選択規則に従ってディジットq_iを決定します．決定されたディジット$q_i \in \{1, 0, \bar{1}\}$の値に応じて，それぞれXレジスタの内容を$\{-X, 0, +X\}$で入力し，現在の部分剰余レジスタ値をもう一方の入力とし，キャリ・セーブ・アダーで演算し，キャリ・セーブ形式の結果を左1ビット分シフトし，次のステップの部分剰余レジスタ値へセットします．このとき，上位3けた分をキャリ伝播させます．開平の場合，Xレジスタには$(Q_i + q_{i+1}2^{-(i+2)})$を設定しますが，$Q_i$はon-the-fly変換が必要です．また，$q_{i+1}2^{-(i+2)}$の余分な項があるので，$Q_i$のon-the-fly変換と絡めて処理する必要があります．

次に，式(13)において開平と逆開平数は，

$$\begin{cases} 除算：R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot \boxed{\left(Q_i + q_{i+1} \cdot 2^{-(i+2)}\right)}\right) \\ 逆開平数：R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot \boxed{X\left(Q_i + q_{i+1} \cdot 2^{-(i+2)}\right)}\right) \end{cases} \cdots(15)$$

であることから，開平と逆開平数の違いは，□で囲った部

図3　on-the-fly変換に使われる2本のレジスタと値の設定のようす

逐次的に決定されるディジットq_iを格納していくPレジスタと，それより一つ小さい値が格納されるNレジスタを設け，$q_i \in \{\bar{1}, 0, 1\}$の決定に応じて，新しく各レジスタに設定する内容を示している．

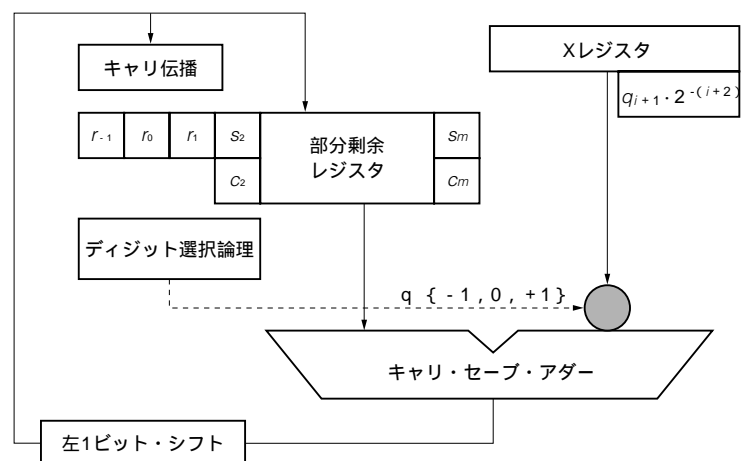
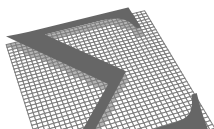



図4　除算と開平演算を共用する演算器のブロック図

除算の除数Xと開平の部分開平値を共通のレジスタ（Xレジスタ）にすることで，除算と開平演算を共用する演算器が設計できる．

分に注目すると，開平の部分項$Q_i + q_{i+1} 2^{-(i+2)}$にXを掛ければ逆開平数の部分項$X(Q_i + q_{i+1} 2^{-(i+2)})$になるということです．そこで部分項$X(Q_i + q_{i+1} 2^{-(i+2)})$を演算する回路を加えることにします．部分項の形から，乗算が必要ですが，

$$Q_i = \sum_{k=0}^{i} q_k \cdot 2^{-k} = Q_{i-1} + q_i \cdot 2^{-i}$$

から，

$$XQ_i = X\left(Q_{i-1} + q_i \cdot 2^{-i}\right)$$

なので，キャリ・セーブ形式で逐次的に$q_i 2^{-i}$を蓄積加算していきます．また，$X \cdot q_{i+1} \cdot 2^{-(i+2)}$ が余分な項として存在しますが，同じように加算していきます．ここで面白いのは，nけたのXの逆開平数を求める場合，最終的に得られるQ_nが逆開平数$1/\sqrt{X}$です．このとき，同時に$X Q_n$を求めていたので，$X/\sqrt{X} = \sqrt{X}$ が得られます．つまり，Xの開平と逆開平数$1/\sqrt{X}$を同時に求めるハードウェア構成が出来上がるということです．

既に，**図4**において，除算と開平を共用するハードウェアを示しているので，ここでもう少し欲張って，除算，開平，逆数と逆開平数で共用できるハードウェア構成を考えてみます．いま，逆数$1/X$を求めたとすると，除算Y/Xは，逆数$1/X$に被除数Yを乗算することで求められます．そこで，

$$\begin{cases} \text{逆数}: R_{i+1} \leftarrow 2(R_i - q_i \cdot X); \quad \text{除算}: \boxed{Q_{i+1} \cdot Y} \\ \text{逆開平数}: R_{i+1} \leftarrow 2\left(R_i - q_{i+1} \cdot \boxed{X\left(Q_i\right.}\left.+ q_{i+1} \cdot 2^{-(i+2)}\right)\right) \end{cases} \cdots(16)$$

のように考えれば，逆数を求めると同時に除算も求まるように，開平と逆開平数で共用できるハードウェア構成が**図5**に示すように出来上がります．**図5**の構成は，**図4**の構成のXレジスタ部分をセレクトできるように拡張しています．主として，XQ_i（逆開平数のとき）またはYQ_i（除算のとき）を演算する回路を設けています．Yレジスタの内容は，Xの開平・逆開平数の演算のときは，Xになります．Zレジスタには，蓄積結果がキャリ・セーブ形式で保存されます．

＊　＊　＊

次回は，今回の採用したディジット選択規則（**表1**）が正しく動作することを検証していく作業を採り上げます．

図5　除算，開平，逆数と逆開平数を共用する演算器のブロック図

除算と開平を共用する演算ブロックに逐次乗算するブロックを付加して，拡張することで，除算，開平，逆数と逆開平数を共用する演算器が設計できる．

参考・引用*文献

(1)　高木直史; 算術演算のVLSIアルゴリズム，コロナ社, 2005.

(2)　Takagi, N., Matsuoka D, and Takagi, K.; Digit-Recurrence Algorithm for Computing Reciprocal Square-Root, IEICE Trans. Fundamentals, Vol. E86, No. 1, pp. 221-228, Jan. 2003.

(3)　島田俊男; 全方位メガピクセルカメラシステム，映像情報インダストリアル臨時増刊号，まるまる! 全方位カメラシステム，pp. 25-32, 2006年4月.

(4)　外村元伸; 拡大・縮小スケーリング・アルゴリズム演算の実装と評価，Design Wave Magazine, 2006年9月号, pp.135-140.

(5)　J.B. Gosling, and C.M.S. Blakeley; Arithmetic unit with integral division and square root, IEE Proceedings, Vol. 34, Pt. E, No. 1, pp.17-23, Jan. 1987.

(6)　J.H.P. Zurawski, and J.B. Gosling; Design of a High-Speed Squre Root Multiply and Divide Unit, IEEE Trans. On Computers, Vol. C-36, No.1 pp.13-23, Jan. 1987.

(7)　松原玄宗，井出進博，鈴木清吾; 非同期回路を用いたハード共用型SRT除算/平方根演算器，電子通信学会　信学技法，DSP95-1-1, ICD95-150, pp. 29-36, 1995.10.

(8)　G. Matsubara, N. Ide, H. Tago, S. Suzuki and N. Goto; 30-ns 55-b Shared Radix 2 Division and Square Root Using a Self-Timed Circuit, Proc. Of 12th IEEE Computer Arithmetic Symp. pp. 98-105, Jul. 1995.

とのむら・もとのぶ
大日本印刷（株）　電子デバイス事業部　電子デバイス研究所

＜筆者プロフィール＞

外村元伸．魚眼レンズは，安価な1万円程度のものと少し高価な4万円程度のもの2本を購入してみました．本格的なものは20万円以上もします．安価な方は今回の写真1で使用しましたがあまり鮮明でないので遊びに使う程度でしょう．現在，少し高価な方を使って魚眼補正処理の実証を行っているところです．